transmission and distribution planning," in *Proc. 4th PSCC* (Grenoble, France, 1972). Paper 1.1/13.
[82] R. A. Evans, "A quadratic programming algorithm for optimal generation planning," presented at IEEE PES Winter Meet., New York, 1973. Paper C 73 112-0.
[83] R. R. Booth, "Power system simulation model based on probability analysis," *IEEE Trans. Power App. Syst.*, vol. PAS-91, pp. 62–69, Jan./Feb. 1972.
[84] ——, "Optimal generation planning considering uncertainty," *IEEE Trans. Power App. Syst.*, vol. PAS-91, pp. 70–77, Jan./Feb. 1972.
[85] A. Breton and M. Falgarone, "Application de la theorie de la commande optimale au probleme du choix des equipments de production a E.D.F," in *Proc. 4th PSCC* (Grenoble, France, 1972). Paper 1.1/7.

Transmission Network Expansion Planning

[86] U. G. Knight, R. R. Booth, S. A. Mallard, and D. M. Lewis, "Computers in power system planning," this issue, pp. 872–883.
[87] W. R. Puntel *et al.*, "An automated method for long-range planning of transmission networks," in *Proc. 8th PICA Conf.* (Minneapolis, Minn., 1973), pp. 38–46.
[88] L. L. Garver, "Transmission network estimation using linear programming," *IEEE Trans. Power App. Syst.*, vol. PAS-89, pp. 1688–1697, Sept./Oct. 1970.
[89] Y. P. Dusonchet and A. H. El Abiad, "Transmission planning using discrete dynamic optimizing," *IEEE Trans. Power App. Syst.*, vol. PAS-92, pp. 1358–1371, July/Aug. 1973.

Distribution Design and Operation

[90] G. Cuervo, G. Navas, and R. Querubin, "Optimal design of radial distribution networks" (in Spanish), presented at Panamerican Cong., Bogota, Colombia, 1973.
[91] R. N. Adams and M. A. Laughton, "A dynamic programming network flow procedure for distribution system planning," in *Proc. 8th PICA Conf.* (Minneapolis, Minn., 1973), pp. 348–354.
[92] K. Matsueda and M. Matsuda, "Secondary bus voltage regulation method at distribution sub-station," in *Proc. 4th PSCC* (Grenoble, France, 1972). Paper 2.1/4.
[93] T. Okada and C. Uenosono, "Mathematical models for optimal design of power distribution systems for long terms," in *Proc. 4th PSCC* (Grenoble, France, 1972). Paper 1.1/3.
[94] J. P. Ludot and M. C. Rubinstein, "Methodes pour la planification a court terme des reseaux de distribution," in *Proc. 4th PSCC* (Grenoble, France, 1972). Paper 1.1/12.

Capacitor Optimization

[95] S. S. Sachdeva and R. Billinton, "Optimal network var planning by nonlinear programming," *IEEE Trans. Power App. Syst.*, vol. PAS-92, pp. 1217–1225, July/Aug. 1973.
[96] ——, "Optimal network var planning using real and reactive power decomposition non-linear analysis," in *Proc. 8th PICA Conf.* (Minneapolis, Minn., 1973), pp. 339–347.
[97] A. Kishore and E. F. Hill, "Static optimization of reactive power sources by the use of sensitivity parameters," *IEEE Trans. Power App. Syst.*, vol. PAS-90, pp. 1166–1173, May/June 1971.
[98] R. M. Porter and H. K. Amchin, "Var allocation—An optimization technique," in *Proc. 7th PICA Conf.* (Boston, Mass., 1971), pp. 310–318.
[99] J. A. Bubenko and B. M. Waern, "A method for the unit commitment of the reactive compensation," in *Proc. 7th PICA Conf.* (Boston, Mass., 1971), pp. 303–309.
[100] A. M. Pretelt, "Automatic allocation of network capacitors," *IEEE Trans. Power App. Syst.*, vol. PAS-90, pp. 54–61, Jan./Feb. 1971.
[101] R. M. Maliszewski, L. L. Garver, and A. J. Wood, "Linear programming as an aid in planning kilovar requirements," *IEEE Trans. Power App. Syst.*, vol. PAS-87, pp. 1963–1968, Dec. 1968

Static State Estimation in Electric Power Systems

FRED C. SCHWEPPE, MEMBER, IEEE, AND EDMUND J. HANDSCHIN, MEMBER, IEEE

Invited Paper

***Abstract*—A static state estimator is a collection of digital computer programs which convert telemetered data into a reliable estimate of the transmission network structure and state by accounting for 1) small random metering-communication errors; 2) uncertainties in system parameter values; 3) bad data due to transients and meter-communication failures; and 4) errors in the network structure due to faulty switch-circuit breaker status information. The overall state estimation process consists of four steps: 1) hypothesize mathematical structure; 2) estimate state vector; 3) detect bad data and/or structure errors; and identify bad data and/or structure errors. The problem is characterized by high dimensionality and the need for real-time solutions using limited computer time and storage. Various methods of solution are discussed and compared.**

Manuscript received January 9, 1974.

F. C. Schweppe is with Electric Power Systems Engineering Laboratory, Energy Laboratory, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Mass. 02139.
E. J. Handschin is with Brown Boveri Research Center, Baden, Switzerland.

I. Introduction

THERE IS NO single structure or list of functions for a modern central control system as each utility or group of utilities has its own special problems and needs. Nevertheless, Fig. 1 shows a "typical" central control system in order to indicate the role of a static state estimator. The many "action oriented" tasks of Fig. 1 work with and from a central data base. Indeed, the life blood of the control system is a base of clean pure data defining the system state and status (voltages, network configuration). This life blood is obtained from the nourishment provided by the measurements gathered from around the system (data acquisition). A static state estimator is the digestive system which removes the impurities from the measurements and converts them into a form which the brain (man or computer) of the central control system can readily use to make "action" decisions on system economy, quality, and security.

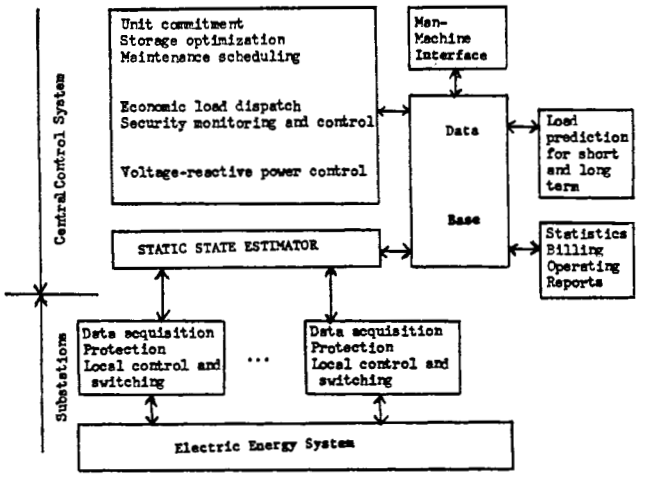

Fig. 1. Schematic diagram for "typical" control center.

There are many ways to design a static state estimator and the choice depends on the particular system's needs. The purpose of this paper is to summarize and compare various possible approaches.

II. Background

The static state estimation concept (as it is being discussed in this paper) has evolved rapidly from its initial inception [1] to on-line implementations; the first is apparently the Tokke system installation in Norway [43] with the larger AEP installation [14] following soon after. Many people have contributed to this evolution as indicated by the papers included in the bibliography. An interesting survey instigated by Ariatti within the CIGRE Study Committee 32, Working Group 01, contains the opinions of more than thirty utilities on on-line power system state estimation [30]. The authors of [14], [20], [26], [37], [42]–[44], [52] discuss static state estimation plans of various companies both in Europe and North America. Other utilities are actively developing a static state estimation capability. However, there was and is no universal acceptance of the concept ([45] is a published critique). There has been some confusion on what a static state estimator is and what its purposes are.

The words "state estimation" are used in modern system (control) theory to refer to a very wide class of problems which are discussed in many papers and books ([54] is one example which at least one of the authors of this paper is prejudiced in favor of). In this general sense, state estimation concepts can be applied in many power system areas ranging from short-range load forecasting to power plant work (see for example [24], [47], [48]). The term "static state estimation" is used in this paper to refer to the particular application indicated in Fig. 1.

Three possible approaches to obtaining the crucial data base of Fig. 1 are 1) data acquisition with no processing; 2) data acquisition with digital computer used for data logging, limit checking, and simple logic comparisons between redundant measurements of essentially the same variable; and 3) data acquisition with a digital computer used in a systematic way, based on a mathematical model, to clean up the data (by treating small random errors, bad data, modeling errors, and parameter errors), and to compute (estimate) quantities and variables which are not directly measured. The third approach is static state estimation. The first approach

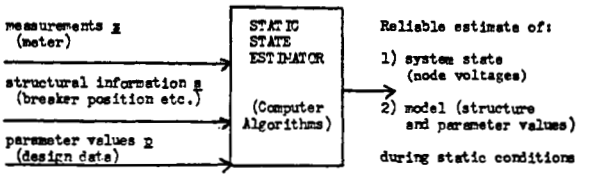

Fig. 2. Purpose of static state estimator.

(data acquisition) simply cannot provide a clean complete data base and so has to be rejected for most central control systems. The second approach (data logging, limit checking, etc.) can provide a partially cleaned up but often still incomplete data base. The third approach (static state estimation) provides a complete data base that is as clean as possible with the given measurement system (or, equivalently, static state estimation requires less metering redundancy to achieve the same level of "cleanliness"). The disadvantage of static state estimation is the increase in digital computer software sophistication and data processing costs.

The preceding "definition" follows the concepts of [1]–[4] by viewing a static state estimator as all the computer codes, logics, and algorithms needed to clean up the data. The fact that static state estimation is based on an overall mathematical model is perhaps its most distinguishing feature.

III. Problem Formulation

A. Definitions and Notation

The purpose of a static state estimator is summarized in Fig. 2. The following definitions provide a basis for the subsequent discussions (numerical values are indicated only to give a feel for the nominal values and many exceptions exist).

Time:

t denotes continuous time; t_n, $n=1, 2, \cdots$, denotes discrete time.

System Conditions:

Transient condition: System behavior immediately following major disturbance. The transient conditions time constant τ_T is the effective time duration of transients. For line switching: $0.1\ \text{s} \leq \tau_T \leq 5$ s; for loss of generators: $2\ \text{s} \leq \tau_T \leq 5$ min.

Static condition: Quasi steady-state system behavior where generators are responding normally to load changes. The static condition time constant τ_S is the amount of time before the system changes an "appreciable" amount: seconds $\lesssim \tau_S \leq$ many minutes.

Independent Variables (Unknown):

Node injections $\mathbf{\Omega}(t)$*:* Real and reactive power injections into nodes (or buses) of the network. Complex, nonstationary, environmentally dependent vector valued stochastic process which is "well behaved" in static conditions. The static condition time constant τ_S is determined by the rate of change of $\mathbf{\Omega}(t)$.

True structure $\xi(t)$*:* Interconnection of transmission lines and transformers and location of meters. Binary type variables; mostly constant except a few changes a day.

True parameter values $\mathbf{p}(t)$*:* Values which determine individual transmission lines, transformers, and meter characteristics exhibit small slow variations with time.

State Variables (Unknown):

State vector $\mathbf{x}(t)$*:* Voltage at all network nodes. During

the static condition it is assumed that

$$\mathbf{\Omega}(t_n) = f[\boldsymbol{x}(t_n), \boldsymbol{\xi}(t_n), \boldsymbol{\varrho}(t_n)] \tag{3.1}$$

where f is a nonlinear function based on Ohm's and Kirchhoff's laws. Dimension $\boldsymbol{x}(t_n)$ is $2N-1$ where N is number of nodes. (N may range anywhere from 10 to 1000.)

Load flow: An algorithm for computing $\boldsymbol{x}(t_n)$ given $\mathbf{\Omega}(t_n)$ $\boldsymbol{\xi}(t_n)$, $\boldsymbol{\varrho}(t_n)$ by solving (3.1) using an iterative technique. (Often some combination of injection and voltage magnitude is specified instead of just $\mathbf{\Omega}$.)

Metering System:

Meter time constant τ_M: $0.1\text{ s} \leq \tau_M \leq 1\text{ s}$.

Meter scan: Operation of reading all meters once and telemetering the data to the central computer. Usually t_n denotes the "time instant" of one scan.

Scan time Δ: Time between two complete meter scans; $\Delta = t_n - t_{n-1}$. $1.0\text{ s} \leq \Delta \leq 30\text{ s}$.

Snapshot scan: All meters are read at the same instant (same value of t between t_{n-1} and t_n).

Sequential scan: Meters are read sequentially at different time instants in the interval $t_{n-1} \leq t \leq t_n$.

Redundancy η: Ratio of number of meters to number of states $1.1 \leq \eta \leq 3.0$.

Local redundancy ϵ_k: Redundancy at node k considers only node k and all other nodes not more than two nodes away from node k.

Lost data: Failure of meter and/or communication link which is known by the central computer.

Known Quantities:

Measurements $\boldsymbol{z}(t_n)$: Telemetered line flows, bus injections, bus voltages, possibly currents, obtained during scan at time t_n. Dimension of $\boldsymbol{z}$ is η times dimension of $\boldsymbol{x}$.

Pseudo measurements: Nontelemetered information on line flows, injections, voltage magnitudes (such as load distribution factors, knowledge of tap changing transformer control logics, etc.). Henceforth, $\boldsymbol{z}$ refers to both telemetered and pseudo measurements.

Parameter values $\boldsymbol{p}(t_n)$: Estimate of $\boldsymbol{\varrho}(t_n)$ obtained from system design data.

Structure $\boldsymbol{s}(t_n)$: Estimate of structure $\boldsymbol{\xi}(t_n)$ obtained from telemetered or telephoned switch and circuit breaker positions and meter status during a scan at time t_n.

Model for Measurements $\boldsymbol{z}(t_n)$:

For a snapshot scan

$$\boldsymbol{z}(t_n) = \boldsymbol{h}[\boldsymbol{x}(t_n), \boldsymbol{\xi}(t_n), \boldsymbol{\varrho}(t_n)] + \boldsymbol{v}_z(t_n) + \boldsymbol{b}(t_n). \tag{3.2}$$

$\boldsymbol{h}[\ \cdot\ ]$ is a nonlinear function based on Ohm's and Kirchhoff's laws. Dimension of $\boldsymbol{z}$ is greater than dimension of $\boldsymbol{x}$.

Measurement error $\boldsymbol{v}_z(t_n)$: Random vector representing the "normal" difference between the actual observation $\boldsymbol{z}(t_n)$ and $\boldsymbol{h}[\ \cdot\ ]$ caused by: 1) nonzero meter time constant, τ_M, combined with noninfinite static condition time constant τ_S; 2) difference between the model $\boldsymbol{h}[\ \cdot\ ]$ in (3.2) and the actual three-phase power system; 3) meter nonlinearities, miscalibrations, and noise; and 4) A/D conversion errors and/or communication noise errors. Time structure of $\boldsymbol{v}_z(t_n)$ is very complex and effectively unknown.

Bias error $\boldsymbol{\beta}$: A major component of measurement errors are biases $\boldsymbol{\beta}$ which are "constant" in time (provided the state itself does not change appreciably). In this paper, biases are modeled as zero mean random variables (constant in time) so that $\boldsymbol{v}_z(t_n)$ is zero mean. However, the bias can be modeled as the unknown mean value of $\boldsymbol{v}_z(t_n)$, in which case it must be estimated (see Section IX).

Measurement error covariance matrix $\boldsymbol{R}$:

$$E[\boldsymbol{v}_z(t_n)\boldsymbol{v}_z^T(t_n)] \triangleq \boldsymbol{R}_{\text{true}}(t_n) \tag{3.3}$$

where "E" denotes expectation. The components of $\boldsymbol{R}_{\text{true}}(t_n)$ are included in $\boldsymbol{\varrho}(t_n)$. (In general, multiplicative errors make $\boldsymbol{R}$ as function of $\boldsymbol{z}$, but this dependence is not indicated to simplify the notation.)

Bad data $\boldsymbol{b}(t_n)$: Unusually large measurement errors added to $\boldsymbol{v}_z(t)$ and caused by: 1) observations during transient swings, 2) major meter-communication failures which are not known by the central computer. $\boldsymbol{b}(t_n)$ is usually zero with large magnitude components occurring with unknown time structure.

Model for Parameter Values $\boldsymbol{p}$:

$$\boldsymbol{p}(t) = \boldsymbol{\varrho}(t_n) + \boldsymbol{v}_p(t_n). \tag{3.4}$$

The parameter value error $\boldsymbol{v}_p(t_n)$ is small with unknown time structure. $\boldsymbol{p}$ contains $\boldsymbol{R}$ which is $\boldsymbol{R}_{\text{true}}$ plus errors.

Model for Structure $\boldsymbol{s}(t_n)$:

$$\boldsymbol{s}(t_n) = \boldsymbol{\xi}(t_n) + \boldsymbol{c}(t_n). \tag{3.5}$$

The structure error $\boldsymbol{c}(t)$ contains binary-type variables which have nonzero components only very rarely. Time structure is completely unknown.

Static State Estimator:

Hypothesize model: Given $\boldsymbol{s}(t_n)$ and $\boldsymbol{\varrho}(t_n)$ determine

$$\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{h}[\boldsymbol{x}, \boldsymbol{\xi}(t_n), \boldsymbol{\varrho}(t_n)]\Big|_{\substack{\boldsymbol{\xi}=\boldsymbol{s} \\ \boldsymbol{\varrho}=\boldsymbol{p}}} \tag{3.6}$$

and the values of $\boldsymbol{R}$.

State estimate $\hat{\boldsymbol{x}}(t_n)$: Value of $\boldsymbol{x}$ which minimizes the residual $\boldsymbol{r}(t_n)$

$$\boldsymbol{r}(t_n) \triangleq \boldsymbol{z}(t_n) - \boldsymbol{h}[\boldsymbol{x}(t_n)] \tag{3.7}$$

in some sense during static conditions.

Estimation: Process of computing $\hat{\boldsymbol{x}}(t_n)$.

Detection: Test to determine whether bad data $\boldsymbol{b}$ and/or structural errors $\boldsymbol{c}$ are present in $\boldsymbol{z}(t_n)$ or $\boldsymbol{s}(t_n)$.

Identification: Process to determine which measurement(s) is (are) bad or which part of the structure is wrong.

Estimate update time, T: Time interval between instants when a new state estimate is computed. $\Delta < T <$ many minutes to hours (Δ is scan time).

B. Overall Approach

The four-step concept of Fig. 3—1) hypothesize structure; 2) estimate; 3) detect; and 4) identify—is an excellent general way to view the static state estimation data processing problem. Fig. 3 applies directly only to the processing of a single measurement scan, but its structure also applies in more general situations. The static state estimator of Fig. 2 includes an "estimator" as shown in Fig. 3.

Sections IV through VIII discuss the four steps of Fig. 3, which handle measurement errors $\boldsymbol{v}_z$, bad data $\boldsymbol{b}$, and most structural errors $\boldsymbol{c}$. Parameter errors $\boldsymbol{v}_p$ are discussed separately in Section IX. Overall system design considerations are discussed in Section X.

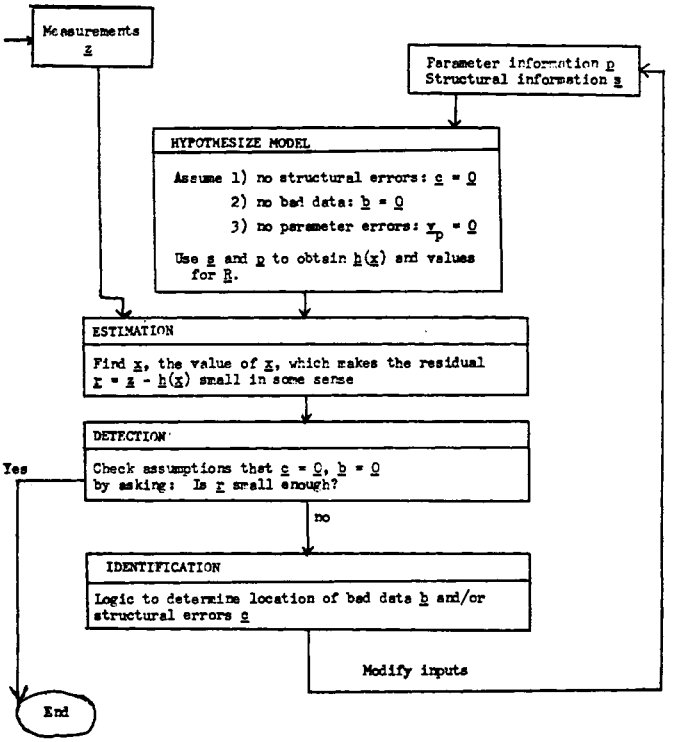

Fig. 3. Four basic operations of static state estimation.

C. Limitations of Paper

It is not possible in a paper of this length to discuss all aspects of static state estimation. A very important aspect of static state estimation is the problem of computer hardware–software organization and the organization of the data base. This is not discussed as it is a complex subject worth a special paper. In actual implementation, a static state estimator would also include prefilters (data screening logics based on limit checking, curve fitting algorithms, etc.) between "Measurements" and "Estimation" in Fig. 3. Such preprocessing methods are not detailed in this paper. In most cases, electric power systems are interconnected to their neighbors so the total system can be divided into two parts, "Own System" and the "Interconnected System" (i.e., the rest of the network). Some interconnected system issues are briefly addressed later in the paper.

IV. Hypothesize Model

Conceptually the role of "Hypothesize Model" in Fig. 3 is very simple. In practice, it is nontrivial to obtain $\boldsymbol{h}(\boldsymbol{x})$ for some given structure $\boldsymbol{s}(t_n)$ as there are a large number of possible switch-circuit breaker–meter location combinations. Careful logical programming based on network theory is required to obtain a useful algorithm. This is an important part of static state estimation, but explicit design techniques are not discussed here. (See [13], [52], [53].)

The state vector $\boldsymbol{x}$ consists of the N complex node voltages which can be expressed either as magnitude and phase angle or real and imaginary components. The choice of coordinate system depends on the application.

The value of $\boldsymbol{R}$ depends on many diverse factors related to the metering system, the network, and the value of the measurements themselves. There is no one set of universal formulas, but the actual computation effort with any approach will be small.

V. Estimation: Time Invariant Case

The second step of Fig. 3 is "Estimation." In this section the estimation using only a single snapshot scan is considered. Dropping the time dependence from the notation, the model is given by

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v}_z \qquad E[\boldsymbol{v}_z \boldsymbol{v}_z^T] = \boldsymbol{R}. \tag{5.1}$$

A. Weighted Least Squares (WLS) Criteria

With the WLS approach [1]–[4], the state estimate $\hat{\boldsymbol{x}}$ is defined as that value of $\boldsymbol{x}$ which minimizes the weighted squared residuals $\boldsymbol{r} = \boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})$; i.e.,

$$J(\boldsymbol{x}) = [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})]^T \boldsymbol{R}^{-1} [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})]. \tag{5.2}$$

WLS estimation is common in many fields and has many good properties. If $\boldsymbol{v}_z$ is Gaussian, then $\hat{\boldsymbol{x}}$ is the maximum likelihood estimate; if $\boldsymbol{v}_z$ is sufficiently small then $\hat{\boldsymbol{x}}$ has a minimum error covariance matrix, etc. (see, for example, [54] for more general discussions).

Differentiation of (5.2) shows that $\boldsymbol{x}$ must satisfy the nonlinear equation

$$\left.\frac{dJ(\boldsymbol{x})}{d\boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \boldsymbol{H}(\hat{\boldsymbol{x}}) \boldsymbol{R}^{-1} [\boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{x}})] = \boldsymbol{0} \tag{5.3}$$

where

$$\boldsymbol{H}(\hat{\boldsymbol{x}}) = d\boldsymbol{h}/d\boldsymbol{x}\,|_{\boldsymbol{x}=\hat{\boldsymbol{x}}}$$

is the Jacobian matrix. In theory there may be several $\boldsymbol{x}$ which satisfy (5.3). However, the multiple-minima problem does not appear important in practice and is ignored in all further discussions.

For an N-node network, (5.3) constitutes a set of $2N-1$ nonlinear equations which must be solved for the $2N-1$ unknowns $\hat{\boldsymbol{x}}$. The equations are "well behaved," so any of a wide variety of iterative algorithms can be used. Various approaches will be discussed.

Minimization of $J(\boldsymbol{x})$ of (5.2) can be viewed as a "sequential unconstrained minimization technique" (SUMT) problem of nonlinear programming [22], [24]. Hence still other types of algorithms can be found in appropriate textbooks on optimization, such as [55].

B. Basic WLS Algorithm

One way to solve (5.3) is the iteration

$$\boldsymbol{G}_i [\boldsymbol{x}_{i+1} - \boldsymbol{x}_i] = \boldsymbol{H}(\boldsymbol{x}_i) \boldsymbol{R}^{-1} [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}_i)], \quad \text{for } i = 0, 1, 2, \cdots \tag{5.4}$$

where $\boldsymbol{x}_i$ is value of $\boldsymbol{x}$ at ith iteration, $\boldsymbol{x}_0$ an initial guess, and $\boldsymbol{G}_i$ is a gain matrix. If convergence occurs, (5.3) is solved as $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i = \hat{\boldsymbol{x}}$, independent of the value of $\boldsymbol{G}_i$ (provided $\boldsymbol{G}_i$ is of full rank). Two issues associated with (5.4) are 1) the choice of gain matrices $\boldsymbol{G}_i$ and 2) the choice of method to solve the linear equation

$$\boldsymbol{G}\boldsymbol{d} = \boldsymbol{b} \tag{5.5}$$

where $\boldsymbol{d} = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i$ and $\boldsymbol{b} = \boldsymbol{H}(\boldsymbol{x}_i) \boldsymbol{R}^{-1} [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}_i)]$.

Linearization with a Taylor series expansion of $\boldsymbol{h}(\hat{\boldsymbol{x}})$ about $\boldsymbol{x}_i$ gives $\boldsymbol{h}(\hat{\boldsymbol{x}}) \approx \boldsymbol{h}(\boldsymbol{x}_i) + \boldsymbol{H}(\boldsymbol{x}_i)[\hat{\boldsymbol{x}} - \boldsymbol{x}_i]$ and substituting into (5.3) assuming $\boldsymbol{H}(\boldsymbol{x}_i) = \boldsymbol{H}(\hat{\boldsymbol{x}})$ yields:

$$\boldsymbol{H}^T(\boldsymbol{x}_i) \boldsymbol{R}^{-1} \boldsymbol{H}(\boldsymbol{x}_i) [\hat{\boldsymbol{x}} - \boldsymbol{x}_i] = \boldsymbol{H}^T(\boldsymbol{x}_i) \boldsymbol{R}^{-1} [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}_i)]. \tag{5.6}$$

This implies that

$$G_i = I_i \triangleq H^T(x_i)R^{-1}H(x_i) \tag{5.7}$$

is a good choice from a convergence point of view. I_i is called the information matrix. Note that G_i in (5.7) depends on x_i.

A direct method to solve (5.5) for d is to use square-root factorization (Choleskey's method) where the gain matrix G is factorized as $G=\sqrt{G}\sqrt{G^T}$ such that $\sqrt{G}$ is a lower triangular matrix (assuming G is positive definite). d is obtained by forward and backward substitution; i.e., $\sqrt{G}y=b$ is solved for y and then $\sqrt{G^T}d=y$ is solved for d. (See, for example, [59], [60].)

The dimensions of z and x can be very large. Thus from a computer implementation point of view, the following facts are important: 1) the Jacobian matrix $H(x)$ is very sparse; 2) elements of $H(x)$ and $h(x)$ have very similar equations so the computations used in obtaining $h(x)$ can be reused to obtain $H(x)$; 3) the information matrix I is sparse (it is roughly "twice as full" as H). Because of these facts, direct implementation is practical for large dimensional problems, provided sparsity programming is used. "Optimal ordering" of the equation improves the effectiveness of the algorithm [64], [65].

For the purpose of this paper the "basic WLS algorithm" is defined as:

> Solve (5.4) with $G_i=I_i$ of (5.7) using square-root factorization, sparsity programming, and optimal ordering. (5.8)

This basic algorithm works very well and can be implemented in a straightforward fashion. Proven computer codes for square-root factorization, etc., are readily available in various languages [59], [60]. The basic algorithm also provides a solid basis for bad data, structural error detection-identification, and for performance analysis, as will be discussed later. However, other algorithms might be better for some applications because of computer time and storage considerations. Some of these alternatives are now discussed.

C. Choice of Gains G

The iteration (5.4) will yield an $\hat{x}$ which satisfies (5.3) for any full rank gain G provided convergence occurs. A linearized stability analysis [5], [54] shows that the iteration tends to be stable (i.e., to converge) for any positive definite G_i such that

$$G_i \geq \tfrac{1}{2}I_i \tag{5.9}$$

so there is a lot of freedom in the choice of G_i. Various approaches will be outlined.

Bandwidth Reduction: The information matrix I is bandwidth limited (provided the nodes are appropriately ordered). Storage requirements can be reduced by using a G with a smaller bandwidth, obtained by prespecifying the desired bandwidth. Any element I_{jk} outside this band is modified as follows: $G_{jj}=I_{jj}+|I_{jk}|$; $G_{kk}=I_{kk}+|I_{jk}|$; and $G_{jk}=0$; while for all other cases $G_{jk}=I_{jk}$. Stability in the sense of (5.9) is maintained.

Constant Gain: A constant gain matrix G such as $G=I(x_0)$ is a very reasonable way to save computer time as $I(x_i)$ does not vary appreciably for the range of x usually of interest [5].

P–θ, Q–V Decoupling: Consider the four variables: P: real power; θ: voltage phase angle; Q: reactive power; V: voltage magnitude. For most operating conditions, the P–θ pair and Q–V pairs are closely coupled with limited cross coupling. By suitable ordering of the variables, a "P–θ and Q–V decomposition" can be used to reduce computer storage by obtaining a block diagonal gain matrix with the "P–V and θ–Q terms" in I equal to zero [4], [40]. Stability in the sense of (5.9) is maintained.

Tuning: The preceding gains are based on modifying the information matrix I. However, this is not necessary. For example, "any" positive definite full rank G can be used where the values of G are "tuned" (adjusted) so that (5.9) is satisfied and good convergence occurs.

In general, the use of $G_i=I_i$ as in the basic algorithm of (5.8) results in the fewest number of iterations but requires the most number of calculations per iteration and the most computer storage. The preceding alternatives require less storage and less computer time per iteration but make possible more iterations.

A true "Newton's method" algorithm results by choosing the gain G_i to be the Hessian matrix $d^2J(x)/dx^2$ [22]. However, the information matrix I_i can be viewed as an approximation to the Hessian matrix where from an implementation point of view, the use of I is superior.

D. Choice of Method to Solve Linear Equations

Direct methods are not the only way to solve the linear equation (5.5). Indirect iterative methods such as Gauss–Seidel, Gauss–Jordan, etc., also exploit the sparsity of matrices [59]. A Gauss–Seidel method will always converge if G is positive definite. Such indirect methods have low computer storage requirements and can be used with any gain.

A direct solution using square-root factorization as in the basic algorithm requires more sophisticated programming and more computer storage. There is no definite answer as to which approach requires the least amount of computer time, but direct methods seem to be preferable.

E. Sequential Algorithms

An iterative technique such as Gauss–Seidel can be applied directly to (5.3). This will "sequentially" try to satisfy each of the $2N-1$ equations in (5.3). Such an algorithm will have close to the smallest possible storage requirements. However, convergence properties and total computer time may not be as favorable.

Debs *et al.* [21] describe an algorithm which can be viewed (roughly) as a combination of an approximation to the information matrix, a "tuning matrix" to obtain convergence, and a Gauss–Seidel iteration. This algorithm processes each measurement sequentially. It has low storage requirements. It evolved from the "simplified dynamic model" approach to be discussed in Section VI-C.

F. P–θ, Q–V Decomposition

The P–θ, Q–V decomposition as discussed in Section V-C for simplifying the gain matrix can also be used to completely separate the state estimation problem into two parts wherein the "θ states" are estimated using only the "P measurements" (see [3]) and the "V-states" using just "V and Q measurements." These separated solutions may be iterated back and forth.

This type of decomposition reduces computer storage requirements. However, it does not actually minimize (5.1) and its accuracy can degenerate in cases of very heavy line loading (power flows).

G. Transformation of Variables ("Line-Only" Algorithm)

The following approach, like "P–θ, Q–V decomposition," is based on the physical nature of the problem, i.e., the structure of $h(x)$ and does not actually minimize (5.1). It is often called the "line-only" algorithm because it was developed to process measurements of line power flows only (see [11]–[16]).

The longitudinal voltage y across a network element is given by the node voltages x and the corresponding line flows z as

$$y = g(x, z). \tag{5.10}$$

The advantage of the nonlinear transformation (5.10) is the property that between y and x the following linear relationship holds: $y = Bx$, where B is the node-element incidence matrix [56]–[58]. The estimation criterion (5.1) is changed to

$$J(x) = [y - Bx]^T R_y^{-1}[y - Bx] \tag{5.11}$$

when y instead of z is interepreted as measurement. Differentiation of (5.11) yields

$$\frac{dJ(x)}{dx} = 2\left[\frac{dy}{dx} - B\right]^T R_y^{-1}[y - Bx] = 0. \tag{5.12}$$

Ignoring the term dy/dx in (5.12) yields with (5.10)

$$[B^T R_y^{-1} B]\hat{x} = B^T R_y^{-1} g(\hat{x}, z).$$

A solution can be obtained with the iteration

$$B^T R_y^{-1} B x_{i+1} = B^T R^{-1} g(x_i, z).$$

It is essential that measurements of lines of different length are weighted appropriately. Therefore, the iteration

$$B^T D B x_{i+1} = B^T D g(x_i, z) \tag{5.13}$$

is used where the elements of the diagonal matrix D are given by $D_{kk} = |\overline{Y}_{kk}|^2$ with $\overline{Y}_{kk}$ the complex admittance of that line on which the kth line flow measurement is taken. The discussion of Section V-D on solving (5.5) applies equally well to (5.13).

One advantage of (5.13) is that $B^T D B$ is independent of the state x as in the "constant gain basic algorithm" of Section V-C (both algorithms require recalculation of gain matrices after a structure change). Another advantage (not apparent from (5.13)) is that estimates for the real and imaginary components of x can be solved separately. (The P–θ, V–Q decomposition algorithm has a similar property.) A potential disadvantage of the line-only algorithm is that there is no guarantee that the resulting $\hat{x}$ will be close to the minimal WLS value; as the criterion is different and the dy/dx term in (5.12) is neglected. The restriction on the types of measurements which can be processed may be a disadvantage; for example, pseudo measurements on load injections cannot be handled. However, the concept can be extended to include other measurements [31], [63]. From the bad data point of view, the basic WLS algorithm offers more flexibility.

H. Weighting Matrices

Another approximate WLS solution is obtained using "weighting" matrices. The basic algorithm of (5.8) can be formally rewritten as

$$x_{i+1} - x_i = W_i[z - h(x_i)] \quad \text{with } W_i = (I_i)^{-1} H(x_i) R^{-1}.$$

An approximate WLS solution can conceptually be obtained by using the iteration

$$x_{i+1} - x_i = W_0[z - h(x_i)].$$

This is not desirable because W_0 is a full matrix. However, many of the elements of W are very small so a practical algorithm is

$$x_{i+1} - x_i = \tilde{W}_0[z - h(x)] \tag{5.14}$$

where $\tilde{W}_0 = W_0$ with small elements set equal to zero [4].

The advantage of (5.14) is that $\tilde{W}_0$ can be computed off-line and the on-line computer time and storage requirements are very small. The disadvantage of (5.14) is that there is no guarantee that the resulting $\hat{x}$ will be close to the minimum WLS value under all conditions. Also, the matrix $\tilde{W}_0$ has to be recomputed after a structure change.

I. Load-Flow Techniques

Consider (5.1) with z partitioned as

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} h_1(x) \\ h_2(x) \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{5.15}$$

where the dimension of z_1 equals the dimension of x. One way to estimate x is to solve the equation [10]

$$z_1 = h_1(x). \tag{5.16}$$

In the case where z_1 corresponds to the bus injections Ω the solution of (5.16) corresponds to the "load-flow" problem for which many very efficient and fast algorithms have been developed [56]–[58]. A common algorithm is the Newton iteration which corresponds to (5.8) when H^{-1} exists. Ways to use the remaining z_2 to improve the estimate by WLS type techniques have been studied [17], [18].

The advantage of this approach is the possibility of using already developed algorithms. A major disadvantage is the difficulty of choosing the subvector z_1 and the effects of errors in z_1 [16]. The approach can be especially vulnerable to bad data and structural errors. The authors of this paper do not recommend this approach.

It is important to distinguish between a load-flow based state estimator (as discussed here) and an "on-line load flow." In this paper, an on-line load flow is defined as a diagnostic tool used to determine the line flows that will result from some hypothesized set of bus injection Ω (or combination of injections and voltage magnitudes) and network structure ξ. Usually, the hypothesized injections correspond to some predicted future condition and/or the hypothesized structure corresponds to some planned or possible network switching. On-line load flows and state estimation serve different functions. On-line load flows were in operation several years before the first state estimator (see, for example, [67]).

J. Spatial Decomposition

The nature of the equation $h(x)$ is such that it is reasonable to spatially decompose (or quantize) z and x into subvectors and then process them individually to "spatially sweep" around the nodes of the network with a recursive algorithm. Because of the nonlinearity several such sweeps may be re-

quired. The decomposition may follow geographical borders of individual utilities, the voltage levels within one power system, etc. (see [4], [25], [39], [40]).

For a single system this has the advantage of reducing the number of equations which have to be solved at any one point in the computation and hence reduces computer storage requirements. However, programming complexity is increased. Computation time advantages or disadvantages are not known. For an interconnected system operation, this spatial decomposition forms the basis for information exchange between control centers.

K. Performance (Error) Analysis

An estimate $\hat{x}$ may not be satisfactory because of an inadequate metering system or too much lost data. Therefore, it is important to have a performance or error analysis which provides a quantitative evaluation of the estimation process.

Such an error analysis is straightforward when the WLS criterion of Section V-A is used. Let x_{true} denote the true but unknown value of x. Equation (5.3) can be written as

$$H(\hat{x})R^{-1}[h(x_{\text{true}}) - h(\hat{x}) + v_Z] = 0.$$

The linearization of $h(x_{\text{true}})$ using a Taylor series expansion about $\hat{x}$ yields

$$\hat{x} - x_{\text{true}} = I^{-1}H(\hat{x})R^{-1}v_Z$$

so that the state estimation error covariance matrix Σ_x, defined as

$$\Sigma_x \triangleq E[(\hat{x} - x_{\text{true}})(\hat{x} - x_{\text{true}})^T]$$

is given approximately by the inverse information matrix I

$$\Sigma_x = I^{-1} = [H^T(\hat{x})R^{-1}H(\hat{x})]^{-1}. \tag{5.17}$$

Usually, one is more interested in the error of some function $y(\hat{x})$ of $\hat{x}$ such as line flows calculated from the states. The error covariance matrix of $\hat{y}$ is approximately:

$$\Sigma_y \triangleq E[(\hat{y} - y_{\text{true}})(\hat{y} - y_{\text{true}})^T] = Y\Sigma_x Y^T \tag{5.18}$$

where $Y = dy/dx$ is the Jacobian matrix of the transformation $y(\hat{x})$ ([54] contains more extensive discussions).

Σ_x is a full symmetric matrix. However, because of sparsity and the nature of the equations, the full Σ_x need not be computed and stored to get Σ_y. Use of square-root factorization of I and sparsity programming makes the on-line computation of Σ_y quite feasible. One of the advantages of the basic algorithm (5.8) is that the extra computer programming effort needed to obtain Σ_y is relatively small.

The residual vector is $\hat{r} \triangleq z - \hat{z} = h(x) + v_z - h(\hat{x}) = Wv_z$ where the residual sensitivity matrix W is given by

$$W = I - H\Sigma_x H^T R^{-1}. \tag{5.19}$$

Performance (error) analysis equations such as the above also can be applied directly to on-line load flows to determine the effect of errors in the hypothesized future injections.

VI. Estimation: Time Varying Case

Section V considered "Estimation" (the second stage of Fig. 3) for the time invariant case. Methods to handle time varying states x and measurement errors v_z are now discussed.

A. Tracking Estimator

Consider a time sequence of snapshot scans so that

$$z(t_n) = h[x(t_n)] + v_z(t_n), \qquad n = 0, 1, 2, \cdots \tag{6.1}$$

with $E[v_z(t_n)v_z^T(t_n)] = R$. If models are not available for the time behavior of either the state $x(t_n)$ or the measurement errors $v_z(t_n)$, the best estimate of $x(t_n)$ will be $\hat{x}(t_n)$ as computed from $z(t_n)$ using the ideas of Section V.

Assume a new estimate is to be computed every scan so that the scan time $\Delta = t_n - t_{n-1}$ equals the estimate update time T.

The iterative solutions of Section V require an initial guess x_0 of the value of x which minimizes $J(x)$. A reasonable initial guess when processing $z(t_n)$ to get $\hat{x}(t_n)$ is to use $\hat{x}(t_{n-1})$. If $T = \Delta = t_n - t_{n-1}$ is small compared to the static condition time constant τ_S, then $\hat{x}(t_n) - \hat{x}(t_{n-1})$ will be small enough so that only one single iteration of (5.4) is required which leads to the tracking algorithm [5]

$$G(t_n)[\hat{x}(t_n) - \hat{x}(t_{n-1})] = H(\hat{x}(t_{n-1}))R^{-1}[z(t_n) - h(\hat{x}(t_{n-1}))]. \tag{6.2}$$

All the discussions of Section V on the choice of the gain matrix $G(t_n)$, methods to solve linear equations, etc., also apply to (6.2). An obvious generalization of (6.2) is to allow the number of iterations per scan to vary as needed.

The simplicity of this tracking concept leads to a straightforward implementation. It is not necessary to make any assumptions on the time structure of either $x(t_n)$ or $v_z(t_n)$. The beauty of not needing to model time structure is paid for by ignoring "information" contained in the past observations.

The preceding concept can be extended to the case where the static state is not updated each scan (i.e., $T > \Delta$). A variety of adaptive logics can be developed to vary the update time depending on how fast the system is changing (i.e., T is a function of τ_S) so that update is done "as needed" [5], [12]. Such logic can result in sizable computer time savings.

B. Sequential Scan

In the case of a sequential scan, all the meters are not read at the same instant of time (simultaneously read meters with sequential data transmission is still a snapshot). The importance of the difference between snapshot and sequential scans depends on the relative values of the static condition time constant τ_S, the meter time constant τ_M, the scan time Δ, etc. For example, if $\tau_S \gg \Delta$ or if $\tau_M \gg \Delta$ a sequential scan is effectively equivalent to a snapshot scan. If there is an important difference, a sequential scan can be treated as a snapshot scan where the difference is accounted for by increasing R by some amount [38].

Situations where it is not reasonable to consider a sequential scan in terms of some equivalent snapshot scan do not seem to occur in practice so further discussions are not given.

C. Simplified Dynamic Models

Valid mathematical models for the time structure of $x(t_n)$ and $v_z(t_n)$ are not available. However, one approach is to use the following simplified dynamic models for the state $x(t_n)$ and measurement error $v_z(t_n)$:

$$x(t_{n+1}) = x(t_n) + w(t_n) \tag{6.3}$$

$$E[\boldsymbol{w}(t_n)\boldsymbol{w}^T(t_m)] = \boldsymbol{Q}\delta_{mn}$$
$$E[\boldsymbol{v}(t_n)\boldsymbol{v}^T(t_m)] = \boldsymbol{R}\delta_{nm} \quad (6.4)$$

where $\delta_{mn}=1$ if $m=n$, $\delta_{mn}=0$ if $m\neq n$; i.e., both $\boldsymbol{w}(t_n)$ and $\boldsymbol{v}_z(t_n)$ are assumed to be white-noise sequences (see [17]–[20]). The assumptions of a dynamic model (6.3) and (6.4) provide, in a mathematical sense, a method to handle sequential scans without viewing them as snapshot scans.

For (6.3) and (6.4) the Kalman–Bucy filtering theory (combined with linearization) provides algorithms for calculating $\hat{\boldsymbol{x}}(t_n)$ where $\hat{\boldsymbol{x}}(t_n)$ now explicitly depends on $\boldsymbol{z}(t_n)$, $\boldsymbol{z}(t_{n-1})$, $\cdots$. Implementation of the exact Kalman–Bucy equations involves the manipulation of full matrices and the choice of $\boldsymbol{Q}$ is not obvious. However, the computer time requirements become practical by diagonalizing the appropriate matrices at each time step. A search for a value of $\boldsymbol{Q}$ which makes the overall algorithm work (tuning the $\boldsymbol{Q}$ matrix), can be done.

The simplified state model (6.3) is at best a very crude representation of the actual state behavior while the simplified observation noise model (6.4) is definitely wrong as much of the measurement noise is "bias" and correlated type errors. Thus it is the personal opinion of the authors of this paper that the basic concept of using a simplified dynamic model and Kalman–Bucy theory is definitely inferior to the concept of combining the tracking ideas of Section VI-A with a suitable algorithm of Section V. However, the resulting algorithm has been successfully implemented. One version of it [21] was discussed in Section V-E for the time-invariant case.

D. Choice of Estimation Algorithm

Consider first the time-invariant case of Section V. Assume a good initial guess is not available and that the iterations are to be continued until convergence occurs. The basic WLS algorithm of (5.8) has the highest storage requirements, while the least storage results either by using a diagonal gain matrix (obtained by bandwidth limiting or tuning) or by applying Gauss–Seidel iteration directly to (5.3). Comparison of computer time requirements is complicated by the tradeoff between time per iteration and number of iterations, but the weighting matrix algorithm of Section V-H can have the smallest on-line time requirements, as the weighting matrices can be completed off line. One measure of the performance of an algorithm is the accuracy of the estimate (assuming no bad data, structural error, or parameter error). The WLS algorithms of Sections V-B–V-E all have the same potential accuracy as they all minimize (5.1). The algorithms of Sections V-F–V-H will tend to give less accurate estimates. However, most present-day applications involve "system security" functions and high accuracy is less important than providing a reliable estimate (i.e., one without undetected large errors). Thus the performance of the detection and identification steps of Fig. 3 (to be discussed in Sections VII and VIII) are more important than accuracy. Since the detection–identification logics to be discussed are based on "residuals" resulting from minimizing (5.1), they are best matched by the basic algorithm of (5.8) or one of the variations of Sections V-C–V-E. The fact that these algorithms are also the most accurate may become important in later years if more accuracy-dependent applications are introduced.

Knowledge of the accuracy of the estimates (or rather line flows) can be very important as certain combinations of lost data, network switching, and injections may cause very large errors to occur in some lines. As discussed in Section V-K, the basic algorithm of (5.8) has the advantage of "automatically" providing such performance (error) analysis.

Now consider the time-varying case and assume the tracking algorithm of (6.2) is used. If the update time is small enough, and if no major system change has occurred, (6.2) can provide satisfactory performance with a simple gain (even diagonal as in [5]). In such a case, use of one of the more sophisticated gains (such as in the basic algorithm of (5.8)) is a waste of computer time.

One reasonable overall approach (suggested in [4]) is to have several different algorithms available where the system operator (or computer itself) chooses the one best suited at any given time. One of many possible combinations is as follows. Code the basic algorithm of (5.8) so that the important performance (error) analysis is available. For iterative solutions after a major system change, use either the basic algorithm or one of its variations where the gain is some approximation to the information matrix. For normal tracking operations use a very simple gain.

VII. Detection

The estimators discussed in Sections V and VI to compute $\hat{\boldsymbol{x}}$ are based on the assumption that no bad data $\boldsymbol{b}$ or structural errors $\boldsymbol{c}$ are present. The third stage of Fig. 3 is "Detection" which tests whether $\boldsymbol{b}$ or $\boldsymbol{c}$ are actually present. This can be viewed as a hypothesis testing problem with the two hypotheses:

$\mathcal{H}_0$: no bad data $\boldsymbol{b}$ or structural errors $\boldsymbol{c}$ are present;
$\mathcal{H}_1$: $\mathcal{H}_0$ is not true.

The following discussions are not as complete as those of Sections V and VI because there exists a "companion" paper [9] which considers detection (and identification) in much more detail and which uses notation and terminology as defined in the present paper. The discussion here is restricted to the time-invariant system as in Section V. The residuals $\hat{\boldsymbol{r}}$ are defined by $\hat{\boldsymbol{r}}=\boldsymbol{z}-\boldsymbol{h}(\hat{\boldsymbol{x}})$ when $\hat{\boldsymbol{x}}$ satisfies (5.3).

One way to test the hypothesis $\mathcal{H}_0$ is the "sum of squared residuals" or "$J(\hat{\boldsymbol{x}})$-test" [3], [4]. If $J(\hat{\boldsymbol{x}})=\hat{\boldsymbol{r}}^T\boldsymbol{R}^{-1}\hat{\boldsymbol{r}}$; then

$$J(\hat{\boldsymbol{x}}) \begin{cases} <\gamma, & \text{accept } \mathcal{H}_0 \\ >\gamma, & \text{reject } \mathcal{H}_0. \end{cases} \quad (7.1)$$

The prespecified parameter γ is the threshold of the test.

A second type of test is the individual residuals $\boldsymbol{r}$ or "$\boldsymbol{r}$-test" using $\tilde{\boldsymbol{r}}=\boldsymbol{D}\hat{\boldsymbol{r}}$ where $\boldsymbol{D}$ is a diagonal weighting matrix. In this test each component $\tilde{r}_k$ of $\tilde{\boldsymbol{r}}$ is considered individually, and if $|\tilde{r}_k|>\gamma$ for some $k=1, 2, \cdots$, then reject $\mathcal{H}_0$. Two different individual residual tests are obtained by using different weighting matrices $\boldsymbol{D}$. The "r_W-test" uses $\boldsymbol{D}=(\text{diag } \boldsymbol{R})^{-1}$. The "$r_N$-test" uses $\boldsymbol{D}=(\text{diag } \boldsymbol{\Sigma}_r)^{-1}$. The r_N-test is superior to the r_W-test but it is harder to compute the weighting matrix $\boldsymbol{D}$.

For both types of tests the choice of the threshold γ determines P_e, the probability of rejecting $\mathcal{H}_0$ when $\mathcal{H}_0$ is true. Assuming $\boldsymbol{v}_z$ is Gaussian (and $\hat{\boldsymbol{r}}$ is small), the distribution of $J(\hat{\boldsymbol{x}})$ and $\hat{\boldsymbol{r}}$ (when $\mathcal{H}_0$ is true) can be evaluated and P_e computed (or bounded) for any γ. In the case of one (or several) bad data, the distribution of $J(\hat{\boldsymbol{x}})$ and $\hat{\boldsymbol{r}}$ can be computed as a function of the size of the bad data. This allows the evaluation of the probability of choosing $\mathcal{H}_1$ (detection) when $\mathcal{H}_1$ is true, i.e., the power function of the previously mentioned tests.

Computer implementation of any of these detection tests is trivial except for the computation of D in the r_N-test which corresponds to doing an error analysis as discussed in Section V-K.

VIII. Identification

If "Detection" indicates the existence of bad data or structural errors, the logic of Fig. 3 proceeds to "Identification" whose function is to decide which measurements contain bad data and/or what part of the structure is wrong. As in Section VII, the discussion here is brief as [9] contains many more details.

In general, it is reasonable to first try to identify bad data and if that fails to try to find structure errors.

One basic approach to identification of bad data is to employ logical search techniques based on the residuals (r_W or r_N). Another seemingly different but actually related approach is to use estimators with nonquadratic error criteria [6], [23]. The computational requirements of either approach can increase the computer time requirements compared to one WLS estimation by a factor of slightly over one to several orders of magnitude depending on the nature and location of the bad data and the meter configuration.

The identification of a structural error is in general more difficult than bad data identification. One approach is to convert the structural error problem into an "equivalent" bad data problem in the parameter $\mathbf{p}$ [7]. Brute force search for structural errors can be aided by the system operator if sufficient computer input–output facilities are provided.

As shown in [9], the effects of bad data are only noticeable in the "neighborhood" of the bad data (this should also occur for the structural errors). It does not spread over the entire network. Therefore, it is possible to decompose the network into a "contaminated" part and a "healthy" part. "Estimation" can continue to update the system state in a tracking mode for the healthy part, while off-line techniques are used to identify the nature of the problem in the contaminated part.

IX. Parameter Errors

The logic of Fig. 3 ignores the difference v_p between the parameter vector $\mathbf{p}$ obtained from system design data and the true value $\boldsymbol{\varrho}$. Such parameter errors are now discussed.

Consider

$$z = h(x, s, \varrho) + v_z, \quad \text{where } \mathbf{p} = \boldsymbol{\varrho} + v_p.$$

Then a Taylor series linearization of $h(x, s, \varrho)$ about $\varrho = p$ yields

$$z = h(x, s, p) + \tilde{v}, \qquad \text{with } \tilde{v} = v_z + \frac{dh}{dp} v_p.$$

If the parameter error v_p is small, its effect can be partially modeled by assuming $\mathbf{p}$ is error-free and increasing the covariance matrix R of the observation error v_z a corresponding amount. It is of course very difficult to decide how much R should be increased but fortunately, the WLS estimation, detection, and identification logics described in this paper tend to be insensitive to errors in R. A paper by Stuart and Herget [35] contains an excellent discussion on the effect of parameter errors with many numerical results (see also [37], [38], [63]).

If there are a few parameters whose values are badly in error it may be possible to estimate them using the identification concepts of Section VIII. Since the parameters are effectively constant in time it is reasonable to use a time series of $z(t_n)$, $n = 1, 2, \cdots$, to get an estimate. Let $\mathbf{p}_1$ denote the subset of $\mathbf{p}$ whose values are to be estimated. The lack of mathematical models for the time structure of $x(t_n)$ and $v_z(t_n)$ can be overcome by defining

$$J(t_n, \mathbf{p}_1) = [z(t_n) - h(\hat{x}(t_n), \mathbf{p}_1)]^T R^{-1} [z(t_n) - h(\hat{x}(t_n), \mathbf{p}_1)] \quad (9.1)$$

where $\hat{x}(t_n)$ minimizes $J(\hat{x}(t_n), \mathbf{p}_1)$ for fixed $\mathbf{p}_1$ and then minimizing

$$J(\mathbf{p}_1) = \sum_{n=1}^{N} J(t_n, \mathbf{p}_1) \quad (9.2)$$

over $\mathbf{p}_1$. Related algorithms can be found in [7], [8], and [62]. One explicit algorithm is to minimize $J(\mathbf{p}_1)$ with respect to $\mathbf{p}_1$ for fixed $\hat{x}(t_n)$, use this new value of $\mathbf{p}_1$ to compute a new set of $\hat{x}(t_n)$, etc. (This is a special case of the algorithm of [61].)

In the case of "interconnected system" operation, it is desirable for each "own system" to have a network equivalent for the outside world. Simulation studies have shown that it is possible that identification of the parameters of the network equivalents (real power) can be computed from the time series $z(t_n)$, $n = 1, 2, \cdots$, of measurements made only in the "own system" [8].

In this paper, $v_z(t_n)$ has been modeled as a zero mean random vector so that the effects of biases β are incorporated in R. An alternative approach (mentioned in Section III) is to consider the biases β to be unknown constants which are to be estimated. With such an approach, the value of R is reduced accordingly, while β is included in $\mathbf{p}_1$, the parameters to be estimated. If biases are viewed as parameter errors which are not estimated, the use of a large R can be viewed as a way to compensate for them. Obviously, the detection tests of Section VII will be more sensitive if the biases are estimated and a smaller R can be used.

X. Overall System Considerations

A. Meter Configuration

In designing the meter configuration one has to guarantee that each component of the state vector x can be estimated. This does not require measurements from all nodes. However, every nonmeasured node has to have an adjacent node where measurements are taken. One type of "minimal cost criterion" is satisfied when all quantities of a selected node are measured because communication links tend to be more expensive than measurement devices. It is reasonable to first select all "important" generator and load buses and then add additional buses as needed. Meter configurations can be compared using the error covariance matrix Σ_x as a criterion [2]. The design of the meter configuration should also consider bad data and structural error detection and identification performance. It is also necessary to consider how lost data affect the degradation of performance. In this context, the design of the corresponding communication system is equally, if not more, important.

The redundancy η alone is not sufficient to describe the quality of the metering system. Reichert *et al.* [25] show how the trace of Σ_x can be reduced by a factor of 4 using different metering configurations having all the same η. Handschin

et al. [9] discuss the fact that the local redundancy ϵ_k is much more important than η for the data detection–identification performance.

B. Implementation

The design of a static state estimator should be tailored to particular needs of the situation and there is no one "best" approach. Designs can range from a basic simple design requiring relatively little computer capacity to highly sophisticated systems. It is also possible to start with an initial simple design which can be "added to" as needed.

The first stage of implementing requires a considerable amount of off-line studies, hardware installations for both computer and telemetering equipment, display techniques, and software development. An initial implementation should be limited to just data processing and display where the estimator serves the operator as an information system and all commands are still given by the operator. Having gained operating experience, more complex functions can be automated such as security monitoring [49], economic load dispatch [50], etc.

XI. Conclusions

On-line static state estimation provides a systematic procedure for building up a reliable and complete data base from raw measurements. It enables the full exploitation of the redundancy of the expensive metering-communication systems at a cost in software sophistication and computer hardware which can be adapted to meet the needs of each individual utility. The concept has already been successfully implemented by several power companies and other installations are in various stages of completion. At the present time, it seems appropriate to classify static state estimation as a proven technique. Even more effective (in terms of computer time-storage) numerical algorithms making full use of the particular nature of the underlying equations should definitely appear (consider, by analogy, the growth in load-flow technology over the years). Computing time and storage requirements will provide one basis for comparing different approaches [36]. There is a definite need to establish some standard "test cases" so that such comparisons can be made.

Acknowledgment

A copy of an earlier version of this paper was sent to many of the individuals who have contributed to the state estimation field. Some very helpful suggestions were received and incorporated into the final paper.

References

(Attempts to provide a complete bibliography rarely succeed. Apologies are made to authors whose work inadvertently was overlooked or misrepresented.)

[1] F. C. Schweppe, J. Wildes, and D. Rom, "Power system static state estimation," Power Syst. Eng. Group, M.I.T. Rep. 10, Nov. 1968.
[2] F. C. Schweppe and J. Wildes, "Power system static-state estimation, part I: Exact model," *IEEE Trans. Power App. Syst.*, vol. PAS-89, pp. 120–125, Jan. 1970.
[3] F. C. Schweppe and D. B. Rom, "Power system static-state estimation, part II: Approximate model," *IEEE Trans. Power App. Syst.*, vol. PAS-89, pp. 125–130, Jan. 1970.
[4] F. C. Schweppe, "Power system static-state estimation, part III: Implementation," *IEEE Trans. Power App. Syst.*, vol. PAS-89, pp. 130–135, Jan. 1970.
[5] R. D. Masiello and F. C. Schweppe, "A tracking static state estimator," *IEEE Trans. Power App. Syst.*, vol. PAS-90, pp. 1025–1033, Mar./Apr. 1971.
[6] H. M. Merrill and F. C. Schweppe, "Bad data suppression in power system static state estimation," *IEEE Trans. Power App. Syst.*, vol. PAS-90, pp. 2718–2725, Nov./Dec. 1971.
[7] ——, "On-line system model error correction," presented at IEEE Winter Power Meet., New York, N. Y., 1973. Paper C 73 106-2.
[8] T. DeVille and F. C. Schweppe, "On line identification of interconnected network equivalents from operating data," presented at IEEE Summer Power Meet., San Francisco, Calif., 1972. Paper C 72 464-6.
[9] E. Handschin, F. C. Schweppe, J. Kohlas, and A. Fiechter, "Bad data analysis for power system state estimation," to be presented at IEEE Summer Power Meet., Anaheim, Calif., 1974.
[10] G. W. Stagg, J. F. Dopazo, O. A. Klitin, and L. S. VanSlyck, "Techniques for the real-time monitoring of power system operations," *IEEE Trans. Power App. Syst.*, vol. PAS-89, pp. 545–555, Apr. 1970.
[11] J. F. Dopazo, O. A. Klitin, G. W. Stagg, and L. S. VanSlyck, "State calculation of power systems from line flow measurements," *IEEE Trans. Power App. Syst.*, vol. PAS-89, pp. 1698–1708, Sept./Oct. 1970.
[12] J. F. Dopazo, O. A. Klitin, and L. S. VanSlyck, "State calculation of power systems from line flow measurements, part II," *IEEE Trans. Power App. Syst.*, vol. PAS-91, pp. 145–151, Jan./Feb. 1972.
[13] A. M. Sasson, S. T. Ehrmann, P. Lynch, and L. S. Van Slyck, "Automatic power system network topology determination," *IEEE Trans. Power App. Syst.*, vol. PAS-92, pp. 610–618, Mar./Apr. 1973.
[14] J. F. Dopazo, S. T. Ehrmann, O. A. Klitin, and A. M. Sasson, "Justification of the AEP real time load flow project," *IEEE Trans. Power App. Syst.*, vol. PAS-92, pp. 1501–1509, Sept./Oct. 1973.
[15] J. F. Dopazo, O. A. Klitin, and A. M. Sasson, "State estimation for power systems: Detection and identification of gross measurement errors," in *Proc. 8th PICA Conf.* (Minneapolis, Minn., June 1973), pp. 313–318.
[16] L. S. Van Slyck and J. F. Dopazo, "Conventional load flow not suited for real time power system monitoring," in *Proc. 8th PICA Conf.* (Minneapolis, Minn., June 1973), pp. 369–376.
[17] R. E. Larson, W. F. Tinney, and J. Peschon, "State estimation in power systems, part I: Theory and feasibility," *IEEE Trans. Power App. Syst.*, vol. PAS-89, pp. 345–352, Mar. 1970.
[18] R. E. Larson, W. F. Tinney, L. P. Hajdu, and D. S. Piercy, "State estimation in power systems, part II: Implementation and applications," *IEEE Trans. Power App. Syst.*, vol. PAS-89, pp. 353–359, Mar. 1970.
[19] A. S. Debs and R. E. Larson, "A dynamic estimator for tracking the state of a power system," *IEEE Trans. Power App. Syst.*, vol. PAS-89, pp. 1670–1678, Sept./Oct. 1970.
[20] A. S. Debs and W. H. Litzenberger, "Experimental evaluation of tracking state estimation," in *Proc. 7th PICA Conf.* (Boston, Mass., 1971).
[21] A. S. Debs, R. F. Larson, and L. P. Hajdu, "On-line sequential state estimation for power systems," presented at 4th PSCC, Grenoble, France, 1972. Paper 3.3/7.
[22] E. Handschin, "Real-time data processing using state estimation in electric power systems," in *Real Time Control of Electric Power Systems*. Amsterdam, The Netherlands: Elsevier, 1972, ch. 2.
[23] J. Kohlas, "On bad data suppression in estimation," *IEEE Trans. Automat. Contr.* (Tech. Notes and Corresp.), vol. AC-17, pp. 827–828, Dec. 1972.
[24] E. Handschin and F. D. Galiana, "Hierarchial state estimation for real-time monitoring of electric power systems," in *Proc. 8th PICA Conf.* (Minneapolis, Minn., June 1973), pp. 304–312.
[25] K. Reichert, E. Handschin, J. Caneel, and H. Asal, "On-line Lastflussberechnung (state estimation) in electrischen Energieversorgungssystemen, Grundlagen," *Elektrotech. Z. Ausg. A*, 1973.
[26] K. Reichert, J. Cancel, H. Asal, and E. Handschin, "Die Ueberwachung eines Teils des Schweizerischen Hochspannungsnetzes mit State-Estimation, Prozessaufbau and praktische Erfahrungen," *SEV-Bulletin* (Special Issue), Zurich, Switzerland, 1973.
[27] E. Handschin and J. Kohlas, "On statistical data processing for power system operation," in *Proc. 4th PSCC* (Grenoble, France, 1972). Paper 3.3/5.
[28] F. Ariatti, "Selection of telemetered quantities for on-line supervision and control of an electric power system with the aid of a real-time computer," presented at Operational Research Symp. (Varna, Bulgaria, May 1970).
[29] F. Ariatti, V. Castagnoli, L. Marzio, and P. Ricci, "Methods for electric power system state estimation: Comparative analysis of computing simulations based upon generalized load flow, tracking and least square methods," in *Proc. 4th PSCC* (Grenoble, France, 1972). Paper 3.3/10.
[30] F. Ariatti, "Answers to questionnaire on on-line power system state estimation," Document of CIGRE Study Committee 32, Working Group 01, 1973.
[31] S. L. Johnson, "An algorithm for state estimation in power systems," in *Proc. 8th PICA Conf.* (Minneapolis, Minn., June, 1973), pp. 319–326.

[32] O. J. M. Smith, "Power system state estimation," *IEEE Trans. Power App. Syst.*, vol. PAS-89, pp. 363–379, Mar. 1970.
[33] E. J. Wolters, "Results of a numerical and a hybrid simulation of a power system tracking state-estimator with an emphasis on line-current measurements," in *Proc. 4th PSCC* (Grenoble, France, 1972). Paper 3.3/3.
[34] A. Z. Gamm, "Power system instantaneous state estimation as a problem in nonlinear programming," *Electr. Technol. USSR*, vol. 3, pp. 89–105, 1972.
[35] T. A. Stuart and C. J. Herget, "A sensitivity analysis of weighted least squares state estimation for power systems," *IEEE Trans. Power App. Syst.*, vol. PAS-92, pp. 1696–1701, Sept./Oct. 1973.
[36] F. Aschmoneit, "Zeit- und Speicherbedarf der exakten on-line Lastflussberechnung," *Elektrotech. Z. Ausg. A*, vol. 94, no. 8, pp. 499–500, 1973.
[37] B. Poretta and R. S. Dhillon, "Performance evaluation of state estimation from line flow measurements on Ontario hydro power system," *IEEE Trans. Power App. Syst.*, vol. PAS-92, pp. 1702–1712, Sept./Oct. 1973.
[38] K. A. Clements, O. J. Denison, and R. J. Ringlee, "The effects of measurement, nonsimultaneity bias, and parameter uncertainty on power system state estimation," in *Proc. 8th PICA Conf.* (Minneapolis, Minn., June 1973), pp. 327–332.
[39] ——, "A multiarea approach to state estimation in power system networks," presented at IEEE Summer Power Meet., San Francisco, Calif., 1972. Paper C72 465-3.
[40] K. Uemura, "State estimation of large scale electric power systems by decomposition methods," in *Proc. 5th IFAC World Congr.* (Paris, France, 1972). Paper 7.5.
[41] J. Bubenko and L. O. Petterson, "State estimation in electric power systems using filtering techniques," in *Proc. 4th PSCC* (Grenoble, France, 1972). Paper 3.3/6.
[42] L. O. Petterson, "Power system state estimation," Dept. Elec. Power Systems Eng., Royal Inst. Technol., Stockholm, Sweden, 1973.
[43] J. Svoen, S. A. Fismen, H. H. Faanes, and A. Johannessen, "The online closed-loop approach for control of generation and overall protection at Tokke power plants," presented at Int. Conf. Large High Tension Electric Systems, CIGRE, Paris, France, 1972. Paper 32-06.
[44] P. Goeminne, P. Van Miegroet, and J. Waha, "Descarte project, a contribution to the improvement of security, in transport networks," in *Proc. 4th PSCC* (Grenoble, France, 1972). Paper 3.3/9.
[45] D. F. Koenig and J. P. O'Malley, "State estimation: An on-line luxury for the electric utilities industry?," presented at IEEE Summer Power Meet., San Francisco, Calif., 1972. Paper C 72 463-8.
[46] H. Kobayashi, S. Ninagawa, K. Tsujiyama, Y. Tamura, and S. Narita, "Dynamic state estimation techniques applied to security control problems," in *Proc. 4th PSCC* (Grenoble, France, 1972). Paper 3.3/2.
[47] F. Galiana and F. Schweppe, "A weather dependent probabilistic model for short term forecasting," presented at IEEE Winter Power Meet., New York, N. Y., 1972. Paper C72 171-2.
[48] R. L. Moore and F. C. Schweppe, "Adaptive coordinated control for nuclear power plant load changes," in *Proc. 8th PICA Conf.* (Minneapolis, Minn., June 1973), pp. 180–186.
[49] F. D. Galiana, F. C. Schweppe, and H. Glavitsch, "Synthesis of security monitoring schemes in power systems," in *Proc. IFAC/IFIP Conf. Digital Computer Applications to Process Control* (Zurich, Switzerland, Mar. 1974).
[50] H. Glavitsch, "Economic load dispatching and corrective rescheduling using on-line information of the system state," in *Proc. 8th PICA Conf.* (Minneapolis, Minn., June 1973), pp. 412–420.
[51] E. Handschin, "Theorie und Anwendung von Estimationsmethoden für die on-line Ueberwachung und Führung electrischer Energiesysteme," Habilitationsschrift, Eidgenössische Tech. Hochsch., Zürich, Switzerland, 1973.
[52] R. G. Rice and G. W. Stagg, "Application program design criteria for the Southern Company control system," in *Proc. 8th PICA Conf.* (Minneapolis, Minn., June 1973).
[53] T. DyLiacco, K. Mamarao, and A. Weiner, "Network status analysis for real time systems," in *Proc. 8th PICA Conf.* (Minneapolis, Minn., June 1973).
[54] F. C. Schweppe, *Uncertain Dynamic Systems*. Englewood Cliffs, N. J.: Prentice-Hall, 1973.
[55] D. A. Pierre, *Optimization Theory with Applications*. New York: Wiley, 1969.
[56] G. W. Stagg and E. H. El-Abiad, *Computer Methods in Power System Analysis*. New York: McGraw-Hill, 1968.
[57] J. R. Neuenswander, *Modern Power Systems*. Scranton, Pa.: International Textbook Co., 1971.
[58] O. I. Elgerd, *Electric Energy Systems Theory*. New York: McGraw-Hill, 1971.
[59] G. Forsythe and C. Mohler, *Computer Solutions of Linear Algebraic Systems*. Englewood Cliffs, N. J.: Prentice-Hall, 1967.
[60] J. H. Wilkinson and C. Reinsch, *Linear Algebra* (handbook for automatic computation, vol. II), F. L. Bauer *et al.*, Eds. New York: Springer, 1971.
[61] R. D. Masiello and F. C. Schweppe, "Multi-machine excitation stabilization via system identification," submitted to IEEE-PAS 1974 Summer Power Meet.
[62] A. S. Debs, "Estimation of steady-state power system model parameters," presented at IEEE-PAS 1974 Winter Power Meet., New York, N. Y., Jan. 1974, Paper T 74 150-9.
[63] K. Srinivasan and Y. Rogichaud, "A dynamic estimator for complex bus voltage determination," presented at IEEE-PAS 1974 Winter Power Meeting, New York, N. Y., Jan. 1975. Paper T 74 153-3.
[64] W. F. Tinney and C. E. Hart, "Power flow solution by Newton's method," *IEEE Trans. Power App. Syst.*, vol. PAS-86, pp. 1449–1460, Nov. 1967.
[65] T. A. Stuart, "Optimal ordering of network buses for weighted least-squares state estimation," presented at IEEE-PAS 1974 Winter Power Meet., New York, N. Y., Jan. 1974. Paper C 74 152–5.
[66] R. Yokoyama, Y. Igarashi, and Y. Tamura, "On detection and location of changes in power system configuration," presented at CIGRE Meet., Study Committee 32, Melbourne, Australia, Mar. 1973.
[67] J. M. Undrill, O. J. Denison, and D. Hayward, "Dispatchers' load flow for the REMVEC dispatch center," presented at IEEE 1969 Summer Power Meet., Dallas, Tex., June 1969. Paper 69 CP660-PWR.